Ce adresa va fi incarcata pe stiva in momentul efectuarii **call**-ului? * 2 puncte

- ○ adresa <proc1>
- ○ adresa <proc1 + 9>
- ○ adresa <main+5>
- ◉ adresa <main+10>

Fie procedura **proc** in care declaram o variabila locala **long t[100]**. Cum va arata golirea stivei astfel incat sa eliminam spatiul alocat pentru **t** pe stiva? * Un punct

- ◉ addl $400, %esp
- ○ addl $100, %esp
- ○ subl $100, %esp
- ○ popl $v

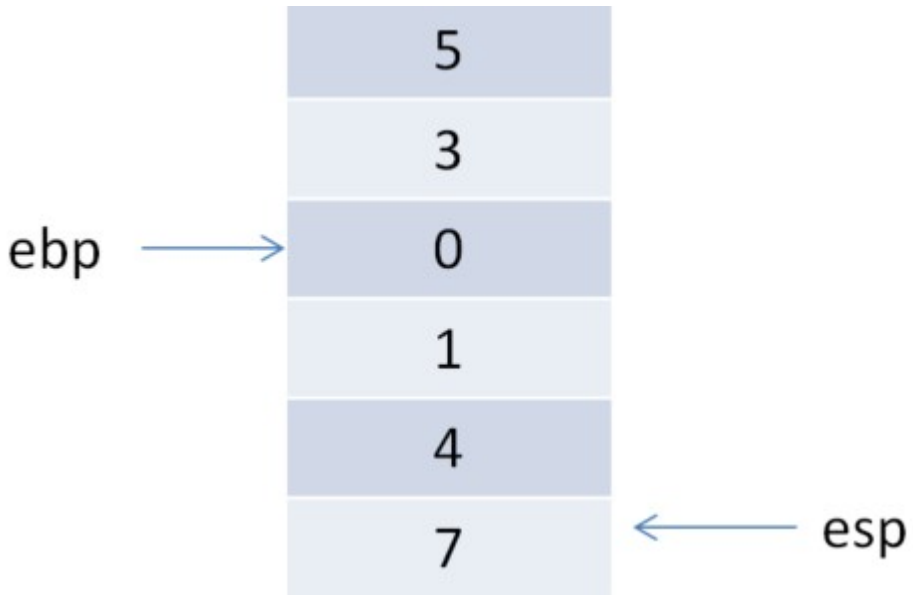
Fie urmatoarele variabile globale: * Un punct

.data
x: .long 5
v: .long 1,2,3,4,5

Ce incarcari pe stiva va produce apelul **proc(x, v[0])**?

- (●) pushl v; pushl x;
- ( ) pushl $v; pushl x;
- ( ) pushl x; pushl $v;
- ( ) pushl x; pushl v;

Cum se realizeaza accesarea lui 5 raportat la **%ebp**? Dar a lui 1? * 2 puncte



- 0(%ebp), 8(%ebp)
- 4(%ebp), -8(%ebp)
- 8(%ebp), -4(%ebp)
- 16(%ebp), 4(%ebp)

Se da codul de mai jos. Raspundeti la urmatoarele intrebari bazate pe acesta.

```
.data
n: .long 3
v: .long 5, 9, 10
nr: .long 0
.text
divizibil_cu_2:
pushl %ebp

movl 8(%ebp), %edi
addl 12(%ebp), %edi

movl %edi, %eax
movl $0, %edx
movl $2, %ebx
divl %ebx
```

```
popl %ebp
ret

.global main
main:

mov $1, %edx
lea v, %esi

subl $1, n

loop:
cmp n, %edx
jg exit
movl (%esi, %edx, 4), %eax
movl -4(%esi, %edx, 4), %ebx

push %eax
push %ebx
call divizibil_cu_2
pop %ebx
pop %ebx

add %eax, nr

inc %edx
jmp loop

exit:
mov $1, %eax
xor %ebx, %ebx
int $0x80
```

In general, accesarile 8(%ebp), respectiv 12(%ebp) indica primul si al doilea argument al unei proceduri. In cazul acesta accesarile argumentelor vor produce **Segmentation Fault.** Indicati instructiunea ce lipseste din cod. * Un punct

- ○ mov %ebp, %esp
- ○ push %esp
- ○ push %ebp
- ◉ mov %esp, %ebp

Ce **registri callee-saved trebuie** salvati pe stiva? Alegeti una sau mai multe variante de raspuns. * Un punct

- ☐ %esi
- ☑ %edi
- ☑ %ebx
- ☐ niciunul

Ce **registri caller-saved trebuie** salvati pe stiva astfel incat in dreptul etichetei **exit** sa fie depozitata in **nr** valoarea corecta (se calculeaza sumele tuturor elementelor consecutive din vector luate 2 cate 2 si se verifica numarul de astfel de sume nedivizibile cu 2) ? Alegeti una sau mai multe variante de raspuns. * Un punct

- ☐ eax
- ☐ ecx
- ☑ edx
- ☐ niciunul

De ce valoarea returnata de procedura nu este folosita corect in cadrul acestui program (procedura returneaza 1 daca suma celor 2 argumente nu este divizibila cu 2 si 0 altfel) ? * Un punct

- ○ Valoarea nu a fost depozitata din procedura nicaieri
- ○ %eax nu a fost salvat pe stiva ca registru caller-saved
- ○ Valoarea returnata a fost alterata dupa iesirea din procedura inainte de a fi folosita
- ◉ Returnarea nu se realizeaza prin registrul corect